**Dialog/Modal**

A modal is an interactive element. It is a popup that appears on top of the current web page.

Generally, a modal forces a user to complete or acknowledge a task and prevents the user from interacting with the page until the modal is closed. For example, to confirm a user is done with filling out a form and ready to submit. Another example of a modal, also known as dialog (<dialog>), is a popup that doesn’t take complete precedence over the page and would allow user to interact with the page while the dialog is open. For example, a popup that displays a privacy policy.

**Usage and Function:** It is an element that is displayed for further interaction and guidance. It could be a message to indicate a successful operation or a form to gather more input from users.

* **Value of component:** 
  + help communicate additional process or respond to user action.
  + communicate the state of a system/process to user.
  + help collect additional input, to continue the process.

**{Component in HTML/CSS and PrimeNG}**

**{Code blocks for use}**

* See attached folder for code.

**When to use**

* Could be used as an alert or a consent form prior to form process or to display intro information needed before completing a form.
* When additional step is required
* When additional information/clarity is needed
* When a processes reaches completion or requires more time
* To display additional information without disrupting a user’s task
* If applicable for usability/accessibility, explain why to use.

It explicitly indicates the initial focus placement by use of the autofocus attribute which help ensure initial focus is set to the element deemed the best initial focus for the dialog.

**When not to use**

* Try to not use modals if possible – try to find a less disruptive component
* When system or process status is intact, and no needed to communicate to user.

(when the process does not need to be interrupted or additional information is not needed)

* Should not replace the main page. Modals are used to focus a user’s attention on a single task or message.
* ~~Implementations details to avoid.~~

~~Use appropriate/applicable popup. Avoid using Modal where Dialog is appropriate.~~

* Processes with multiple steps
* Error, success, and warning messages – instead, include the messages within the context of the page’s content. For example: a error message with the question as part of the validation. Use an alert banner for page-level communication.

**Accessibility for {modals}**

The native HTML <dialog> element should be used in creating modal dialogs as it provides usability and accessibility features. Although, <dialog> element itself would provide the best initial focus placement, It is important to consider the most appropriate place to set user focus if needed. Explicitly indicating the initial focus placement by use of the autofocus attribute will help ensure initial focus is set to the element deemed the best initial focus placement for any particular dialog

Use the appropriate .showModal() or .show() method to render dialogs. When using HTMLDialogElement.showModal() to open a <dialog>, focus is set on the first nested focusable element.

device with a keyboard, the Escape key is commonly expected to close modal dialogs as well. By default, a <dialog> invoked by the showModal() method will allow for its dismissal by the Escape. A non-modal dialog does not dismiss via the Escape key by default, and depending on what the non-modal dialog represents, it may not be desired for this behavior. If multiple modal dialogs are open, Escape should only close the last shown dialog. When using <dialog>, this behavior is provided by the browser.

The <dialog> element is exposed by browsers similarly to custom dialogs using the ARIA role="dialog" attribute. <dialog> elements invoked by the showModal() method will have an implicit aria-modal="true", whereas <dialog> elements invoked by the show() method, or rendered by use of the open attribute or changing the default display of a <dialog> will be exposed as [aria-modal="false"]. When implementing modal dialogs, everything other than the <dialog> and its contents should be rendered inert using the inert attribute. When using <dialog> along with the HTMLDialogElement.showModal() method, this behavior is provided by the browser.

**Best practices for {Modal}**

Users should trigger modals. Modals should appear as a result of an action made by the user or (less commonly) inactivity. A modal should not surprise the user, so don’t automatically display them. Some exceptions would be to alert the user that their authenticated session in a web application is about to expire due to inactivity or if information needs to be displayed when arriving at a webpage for the first time (like accepting the use of cookies).

Choose the modal size that fits your content. There are two sizes to choose from: default and large. If a modal’s content can’t fit without scrolling, you may need to use the large variant or consider moving the content to its own page.

Use clear header and button text. The header should clearly state what’s happening or what action the modal is prompting the user to do. The button text should indicate what will happen when selected and avoid ambiguity. For example, if a modal asks “Do you want to cancel?” the button options should not be “yes” and “cancel” — instead use something like “Yes, cancel” and “No, don’t cancel.”

Limit in-modal interactions. Avoid using components other than buttons inside the modal. Components such as accordions and form fields often don’t scale well for mobile users and they can easily lose context that they’re viewing a modal. If you need to include links that navigate away from a modal to another webpage, refer to our link guidance about how to handle external links.

Consider what happens when the modal is dismissed. The page underneath should not reload or change to new content.

Avoid long content that requires scrolling. If a lot of content is needed, make sure it’s clear that users have to scroll to see all of it. Lengthy content can be problematic because it pushes buttons out of a user’s initial view, which may cause confusion.

Don’t roadblock external links with a modal window or dialog box. Allow users to follow external links without taking a separate action to acknowledge leaving your site. Roadblock notices result in a poor user experience and are redundant with both the link’s destination context and your site’s policies and notices page (see Link guidance).